



SRv6 Powered Backbone Networks

Jasmeet Bagga *Software Engineer, Meta*

Andrey Golovanov *Network Engineer, Meta*



OCP
PLATINUM
Member



OCP
GLOBAL
SUMMIT

2026

Scaling Innovation for the AI Era.

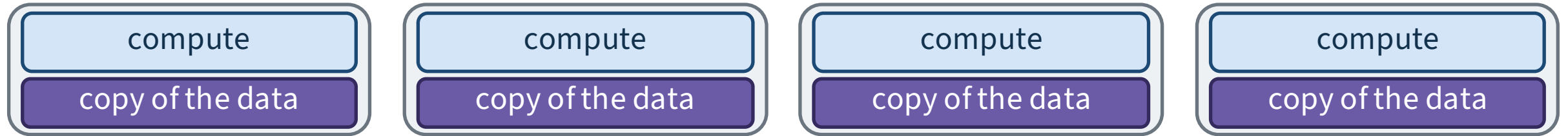
Agenda

- 1 Why Rings, and RBB** services shared inside a latency envelope
- 2 Shallow buffers at 650 km** what ECN makes possible
- 3 Why SRv6** we run MPLS SR TE at scale and still chose the other one
- 4 How we use it** data plane and control plane
- 5 Design to production** and what SAI still needs

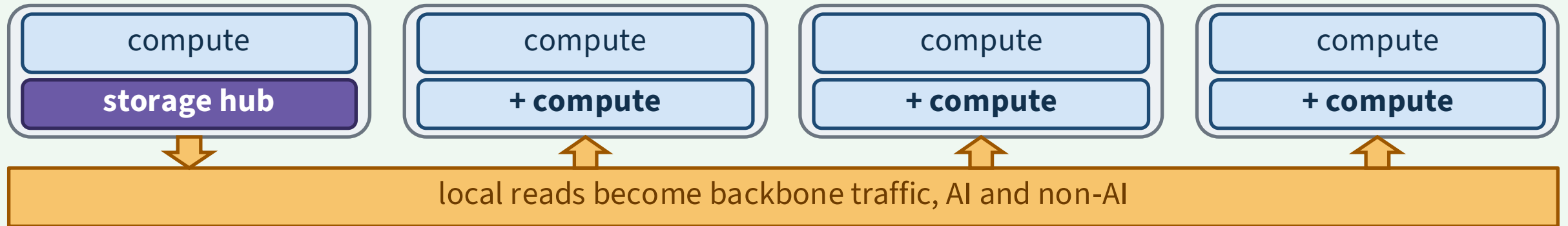


Why Rings

Today — compute must read data in its own region



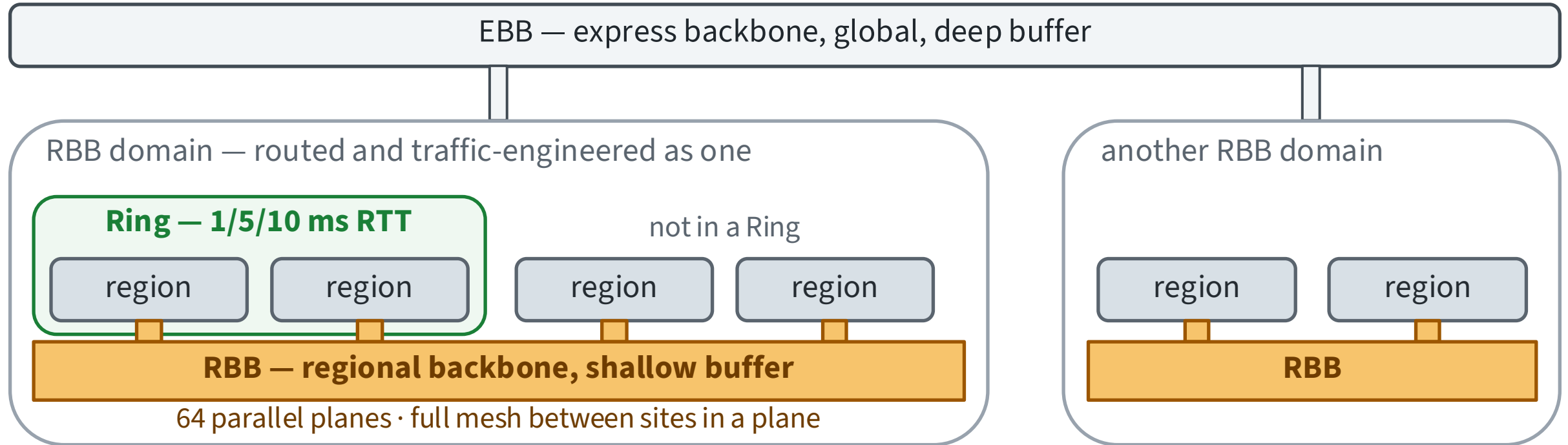
In a Ring — same space and power, one shared pool, not four copies



Copies become compute. The network carries the difference.

RBB: the network that carries it

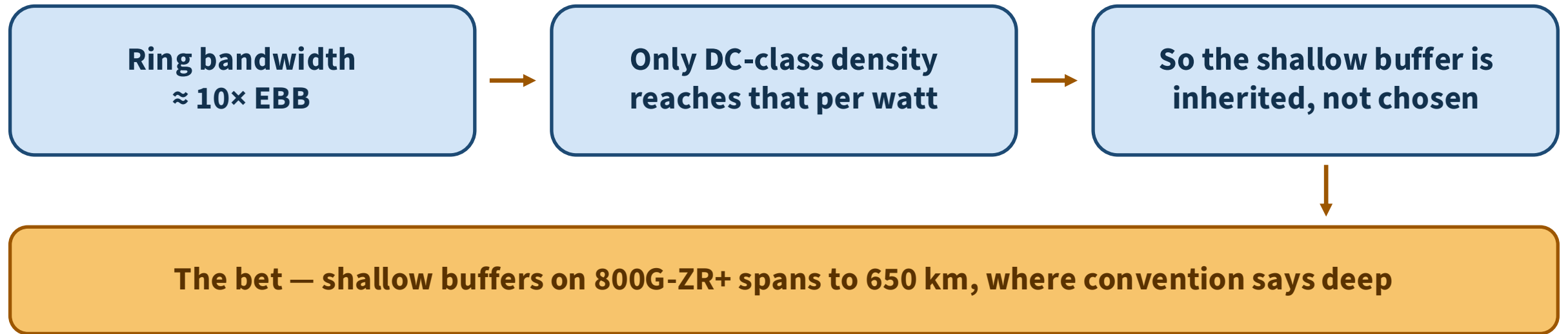
A regional backbone — demand is local to domain; capacity scales with space and power



Same-domain traffic moves from EBB to RBB.

Shallow buffers at 650+ km

Why shallow — a consequence of the efficiency requirements



Requirements chose the platform.

ECN makes shallow-buffer long haul viable

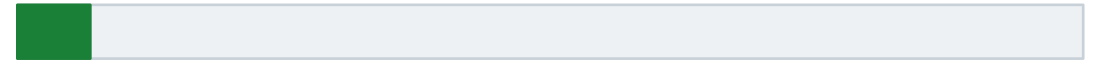
GLIDE backs senders off on the ECN mark — ours, measured at 5 ms one-way



70%

of the link

target reached with GLIDE



~7%

of the buffer

base, common case

A 4.5-to-1 incast is what fills it — 32 to 53 MB.

Most of the link. Almost none of the buffer.

Why SRv6

We run MPLS SR TE at scale and still chose SRv6

Olympic QoS needs six classes, strict priority



MPLS Traffic Class

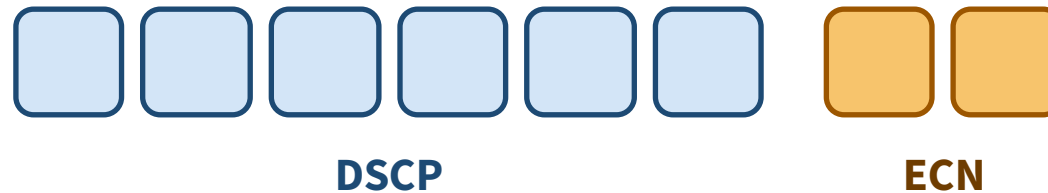
3 bits, RFC 5129



4 classes

IPv6 Traffic Class

8 bits, RFC 6040



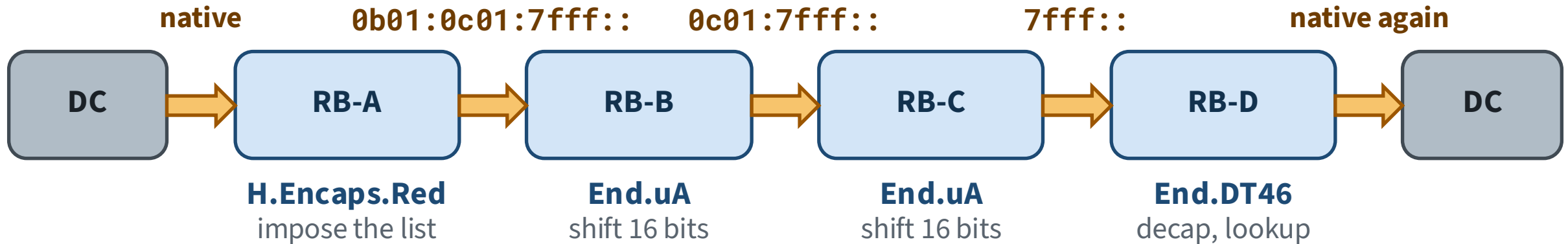
all six

Six needed. Our MPLS encoding gives four.

How we use SRv6: the data plane

One address, six slots — each hop consumes one and forwards

outer destination, after the fcbb:bb00 locator block:



mySID table at RB-B, programmed once at bootstrap

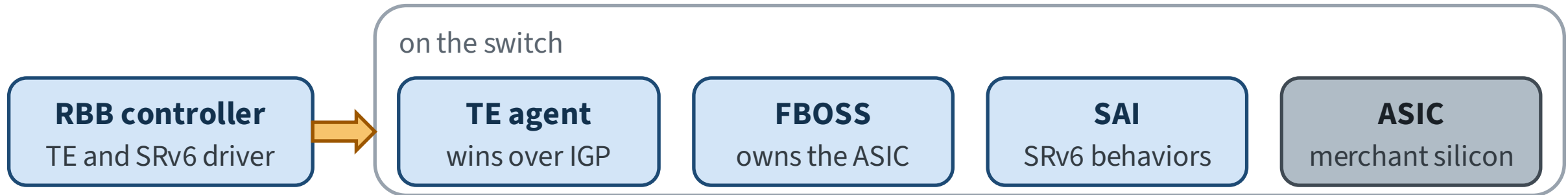
fcbb:bb00:0b01::/48 ADJACENCY_MICRO_SID → Port-Channel to RB-C

End.B6.Encaps.Red stitches two lists when a path needs more than six

One entry per adjacency. Not one per path.

How we use SRv6: the control plane

One controller per RBB plane programs every switch in five calls, every 30 seconds



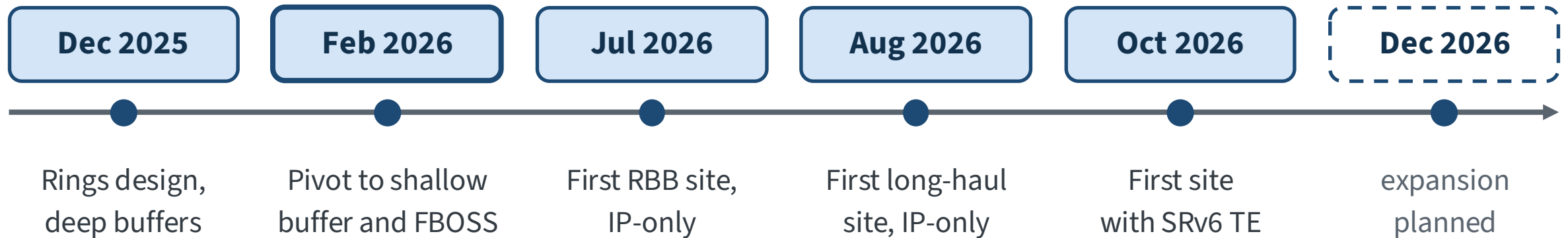
`addMySidEntries()` the mySID table on every RB — node and adjacency SIDs
`addOrUpdateNamedNextHopGroups()` the head-end's set of paths — all or nothing
`addUnicastRoutes()` the route that points at that group
`addPolicies()` a next-hop group per forwarding class
`addAdjacencyFrr()` backup next hops for a protected uA mySID

The controller optimizes. FRR sits below it.

Design to production

Seven months — deep-buffer design to shallow-buffer production

the decision this talk is about



First RBB sites on SAI with Cisco Silicon One. More platforms to follow

Five months from the pivot to the first RBB site.

The ask: what the stack still needs

The core spec holds up — the gaps are at the edges

Base SRv6 behaviors: specified, implemented, confirmed on multiple ASICs

In SAI, FRR and PBR are less well specified

FRR — no definition of a monitored object down

Class-based forwarding — no support

Two more asks — tell us how you build long haul, and share what SRv6 taught you

Help us evolve SAI for TE.

Thank You!



OCP
GLOBAL
SUMMIT

October 12-15, 2026
San Jose, California